

Scene Modeling and Augmented Virtuality Interface for Telerobotic Satellite Servicing

Balazs P. Vagvolgyi¹, Will Pryor¹, Ryan Reedy^{1,2}, Wenlong Niu^{1,3}, Anton Deguet¹,
Louis L. Whitcomb⁴, Simon Leonard¹ and Peter Kazanzides¹

Abstract—Teleoperation in extreme environments can be hindered by limitations in telemetry and in operator perception of the remote environment. Often, the primary mode of perception is visual feedback from remote cameras, which do not always provide suitable views and are subject to telemetry delays. To address these challenges, we propose to build a model of the remote environment and provide an augmented virtuality visualization system that augments the model with projections of real camera images. The approach is demonstrated in a satellite servicing scenario, with a multi-second round-trip telemetry delay between the operator on Earth and the satellite on orbit. The scene modeling enables both virtual fixtures to assist the human operator and augmented virtuality visualization that allows the operator to teleoperate a virtual robot from a convenient virtual viewpoint, with the delayed camera images projected onto the 3D model. Experiments on a ground-based telerobotic platform, with software-created telemetry delays, indicate that the proposed method leads to better teleoperation performance with 30% better blade alignment and 50% reduction in task execution time compared to the baseline case where visualization is restricted to the available camera views.

I. INTRODUCTION

Teleoperation enables a human operator to perform operations in inaccessible or remote locations, including hazardous or extreme environments such as zones of high radiation, the depths of the ocean, and outer space. There are, however, several potential challenges to obtaining effective teleoperation performance. In this study, we consider two of these challenges, which are: (1) the latency of the communication channel between the master control console and the remote robot, and (2) the visibility of the remote environment provided by the available sensors. In particular, for remote environments such as the deep ocean and outer space, it is extremely challenging or impossible to obtain high quality communications and visualization. One example is teleoperation for servicing of satellites on orbit, which motivated this study.

Most satellites on Earth orbit are designed with a finite service life limited by on-board consumables—principally fuel for orbital maneuvering and attitude control. Moreover, if a fault occurs on a satellite on-orbit, we presently lack the ability to conduct on-orbit repairs. To address this deficiency in servicing capability, the Satellite Servicing Projects Division

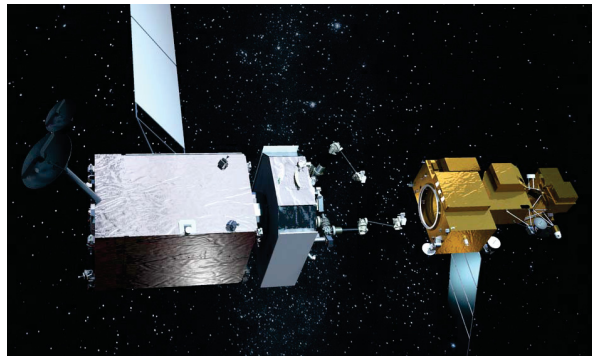


Fig. 1. Artist's concept of Restore-L from NASA's website [15]. Servicing spacecraft (left) approaching Landsat 7 (right).

at the NASA Goddard Space Flight Center is investigating the use of remotely-controlled robotic spacecraft for servicing satellites on orbit. In May 2016, NASA announced the Restore-L mission (Fig. 1) to demonstrate telerobotic refueling of Landsat 7, a U.S. government-owned Earth-observation satellite on low-earth orbit.

The concept for Restore-L is to fly a servicing spacecraft with two robotic arms that can be teleoperated from the ground, with round-trip delays between 2 and 7 seconds. Once the servicing spacecraft is autonomously docked with the satellite, repair operations would be performed telerobotically. In this study, we focus on one of several difficult tasks arising in this mission: the task of gaining access to the fuel fill valves by cutting through the multi-layer insulation (MLI) that covers the satellite. On Landsat 7, these valves are covered by an MLI “hat”, which has a rectangular geometry but is not structurally rigid. Thus, it is necessary to cut through the MLI hat on three of its sides; based on the anticipated camera configuration and geometry of Landsat 7, one of these sides must be cut without any direct camera visualization. NASA has determined that the MLI cutting task is far too complicated to be fully automated and therefore adopted a teleoperated approach, but payload constraints and reliability concerns preclude the use of additional cameras or actuated degrees of freedom for the existing cameras.

The Restore-L mission requires methods to compensate for the limitations in the quality of the communications channel and the remote visualization. To address the challenge of communications delay, we previously developed a model-based approach [23], [24], following the model mediated telemanipulation concept [14], and demonstrated it in multi-user ground-based experiments [21], [22]. For the visualiza-

¹Dept. of Computer Science, Johns Hopkins University, Baltimore USA, {balazs, willpryor, sleonard, pkaz}@jhu.edu

²Dept. of Mechanical Engineering, University of Central Florida, USA

³National Space Science Center, Chinese Academy of Sciences, China

⁴Dept. of Mechanical Engineering, Johns Hopkins University, Baltimore USA, llw@jhu.edu

tion challenge, we recently reported an *augmented virtuality* interface [20], where the operator interacted with a *virtual* 3D model of the satellite that was augmented by projecting the *real* images from the robot tool camera onto the satellite model. This model-based approach was demonstrated, without time delay, on a task that was representative of cutting the MLI; specifically, using a marker to draw lines on the MLI surface.

The results reported herein extend these prior results by: (1) updating the 3D model with reconstructions of “unknown” objects (i.e., objects not in the satellite CAD model), such as the MLI hat, (2) allowing the operator to plan the desired cutting path on the reconstructed hat, (3) providing motion assistance, such as virtual fixtures [16], on the teleoperation master console, and (4) evaluating the augmented virtuality approach for a task that more closely emulates cutting the MLI hat, subject to a telemetry time delay of 5 seconds between master and slave, and comparing to the conventional visualization approach (i.e., delayed feedback from a selected camera). We note that a major advantage of the model-based augmented virtuality approach is that it provides visualization even when performing manipulation tasks in “blind spots” where the remote camera views are obstructed.

II. RELATED WORK

The research of teleoperation with time delay can be traced back to the 1960s, when Ferrell observed that the presence of significant time delay can negatively affect an operator’s performance in completing a task and proposed a “move-and-wait” strategy to avoid stability issues [5]. Recent studies have shown that delayed feedback results in degraded performance [1] and increased operator frustration [25]. For systems with several seconds of delay, one early effort was predictive display [2], [3], where the system predicts and displays the future position of the robot, often as an augmented reality overlay on the delayed camera images. Another approach is supervisory control [17] where, instead of directly teleoperating the remote robot, the operator issues high-level goal specifications, supervises the execution process and intervenes when errors occur. Model-based methods have increasingly been adopted for time-delayed teleoperation [6], [7], [14], [26]. This includes teleprogramming [6] and tele-sensor-programming [7], which allow the operator to interact with a simulated remote environment and teleprogram the remote robot through a sequence of elementary motion commands. Model-mediated telemanipulation [14] creates a model from the slave sensor data; this model is rendered haptically to the operator without delay. On the remote side, the robot controller only executes the position/force commands from the master that are consistent with the model.

Separately, the challenge of teleoperating with limited camera viewpoints has been studied. One prior study demonstrated that the limited selection of camera perspectives available during conventional space teleoperation poses a significant mental workload [12]. In the medical domain, researchers have reported approaches for visualizing endoscopic camera images from alternate viewpoints [10], [11]. Draelos et

al. [4] recently presented an Arbitrary Viewpoint Robot Manipulation (AVRM) framework, targeted at visualization of 3D optical coherence tomography (OCT) imaging during ophthalmic surgery. These prior approaches offer visualization of real-time images from alternate viewpoints and therefore cannot provide visualization when cutting blind spots, as is possible with our model-based approach.

The two most recognized mixed reality concepts are augmented reality (AR) and augmented virtuality (AV) [13]. Both combine visual representations of real and virtual environments. In AR, virtual objects are overlaid on video streams. Since the 1990s, NASA has been experimenting with AR in teleoperation while servicing the ISS and other satellites to improve the operators’ situational awareness [8].

In contrast, in augmented virtuality (AV) the result is a computer generated rendering of the environment, in which registered real-life images are overlaid on virtual objects. This approach enables visualization from arbitrary points of view, as opposed to AR, where the location of the camera is fixed. AV also enables the rendering of stereoscopic views of the scene, which has been shown to improve teleoperation performance [18].

III. TELEOPERATION SYSTEM DESCRIPTION

The major components of the teleoperation system are depicted in Fig. 2, with conceptual contributions represented by the Scene Modeling and Augmented Virtuality Visualization components. Implementation details for these two components and the overall teleoperation system are provided in the following subsections.

A. Scene Modeling

Scene Modeling is provided by the Vision Assistant software, which is responsible for image capture, image calibration, hand-eye calibration, registration, modeling of unknown objects, definition of cutting paths (virtual fixtures), and simulation of time delay. This consists of a small set of C++ applications with Qt GUI and ROS interfaces, and is deployed on the computer that handles video capture from the cameras. In an actual servicing mission, video capture would occur on-orbit and the remaining functions would be performed on the ground, as shown in Fig. 2.

1) *Calibration and Registration:* Accurate augmented virtuality visualization requires precise registration between the CAD model and the real camera images of the client satellite. This registration can be calculated by locating the satellite’s natural landmarks within the images, then using pose estimation to find the satellite pose that best fits these observations. If the camera’s pose (extrinsic parameters) is known from robot kinematics, then the satellite’s pose with respect to the camera will also yield a registration of the satellite to the robot’s base frame. Pose estimation is sensitive to the landmark observation accuracy; thus, we combine pose estimates from multiple camera viewpoints to obtain more accurate registration.

This registration procedure requires the camera’s extrinsic and intrinsic parameters. The camera intrinsics can be

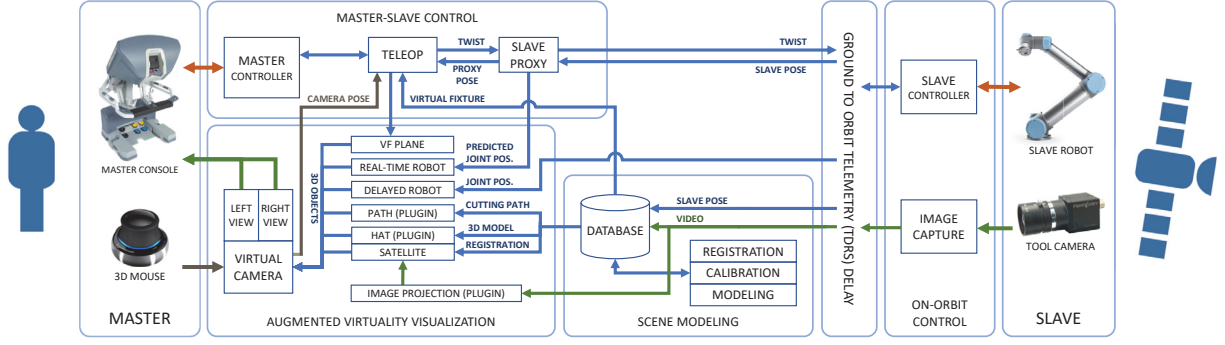


Fig. 2. System diagram of the augmented reality teleoperation system. The telemetry delay was simulated in software, with no upstream delay and 5 seconds downstream delay. The Slave Proxy, simulated delay, and Slave Controller were combined in a single component for ease of implementation.

calibrated prior to launch, and they are unlikely to change during the mission. However, the Vision Assistant is capable of re-calibrating the camera during flight using a checkerboard pattern or natural landmarks. The Vision Assistant can also calculate the tool camera's extrinsic parameters using either natural features or a checkerboard pattern. The algorithm first uses Tsai's method [19] to solve the conventional $AX=XB$ hand-eye formulation, then refines X using reprojection error minimization.

2) *Modeling Unknown Geometry*: The MLI hat is a soft structure that is not included in the CAD model of the satellite and therefore must be modeled based on an image survey performed with the tool camera. The modeling takes place in the Vision Assistant, where the operator manually locates natural landmarks on the MLI that are unambiguously identifiable on at least two images taken from different view angles. Once the landmark observations are added, the software automatically calculates the landmark positions in 3D space with respect to the satellite's base coordinate frame. The triangulation algorithm uses a closed-form least squares method to find the best positions given at least two observations per landmark.

Knowing the 3D coordinates of the MLI landmarks enables the user to create triangular or quadrilateral 'faces' between the landmarks and build a model of the MLI object. The landmarks serve as vertices and the faces are converted into triangles that form the topology of the mesh. The Vision Assistant sends the mesh to the visualization computer, where a custom RViz plugin converts it into an Ogre3D object and adds it to the OpenGL scene (Fig. 3).

While we only modeled the box-like MLI hat for our specific application, the method allows the construction of arbitrary 3D shapes – e.g. a deformed MLI hat – as long as 3D landmarks marking the shape's outlines are available.

3) *Definition of Cutting Path*: The Vision Assistant provides the capability to define paths by connecting multiple landmarks with one continuous line. However, the desired cutting path may not be located between uniquely identifiable natural landmarks. In fact, for the MLI cutting task, the path lies on a flat, featureless area, where picking a landmark on multiple camera views would be nearly impossible, especially considering the highly reflective nature of the MLI. For

this reason, the Vision Assistant provides a helper tool for placing landmarks with high precision in such featureless spots. The tool enables the user to project a metric grid on any previously modeled quadrilateral face. The resolution of the grid can be customized to arbitrary precision, and the coordinate frame of the grid can be aligned with any corner and side of the face. Using the grid tool, the user can easily pick any point on any previously modeled surface with sub-millimeter precision, then create a 3D landmark on that grid point. The quadrilaterals defined by the user need not be planar (i.e., skew quadrilaterals), therefore the projected grid can follow the 3D curvature of the faces.

In our experiments, the ideal cutting path, defined in the Vision Assistant, was displayed for the robot operators in the master console (e.g., purple line in Figs. 3 and 4). For robot control, however, we chose to represent each line in the path by a virtual fixture plane, so that the operator can easily control the depth of cutter penetration into the MLI. For the experiments reported here, all cut path lines were designed to lie in a single plane, so we defined the virtual fixture by fitting a plane to the landmarks of the ideal cutting path.

B. Augmented Reality Visualization System

The master side visualization system is responsible for rendering the stereoscopic augmented reality view for the master console. It comprises a collection of ROS RViz plugins and configuration files, and is capable of rendering the mixed reality scene featuring the following elements (Fig. 4):

Client satellite: A combination of known geometry (satellite CAD model) and reconstructed geometry (MLI hat).

Time-delayed robot, camera, and cutting tool: Represents the state of the robot acquired through delayed telemetry. Rendered semi-transparently.

Real-time robot, camera, and cutting tool: The command the robot operator is sending to the satellite.

Delayed video projection: Camera images arrive after a few seconds of delay. Rendered with high opacity.

Ideal cutting path and virtual fixture: Cutting path is modeled in Vision Assistant and virtual fixture is calculated from points on the path.

The visualization module is also responsible for rendering the conventional teleoperation display used for the baseline experiments. In this mode, the master console displays the

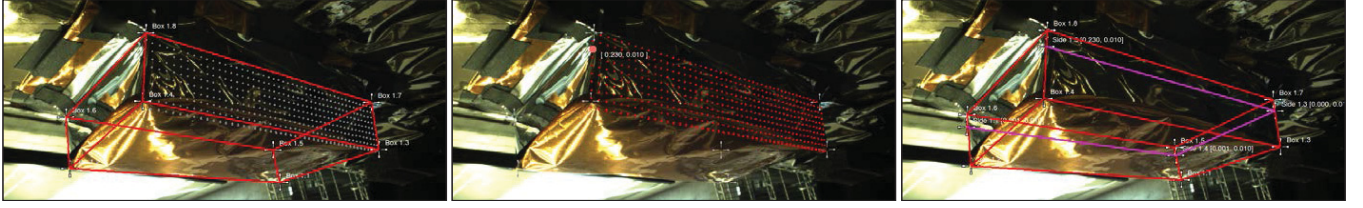


Fig. 3. Modeling of MLI hat and the cutting path in Vision Assistant. Geometry of the hat shown with a 5 mm grid on one side (left), user selects grid point for landmark creation (center), cutting path defined and shown as a purple line (right).

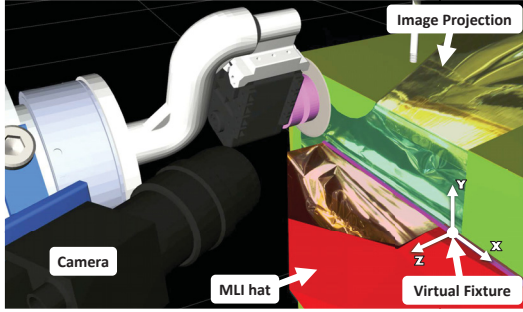


Fig. 4. Augmented virtuality view during cutting (labels and coordinate frame added for clarity); see Fig. 8 for a real image of a similar scene.

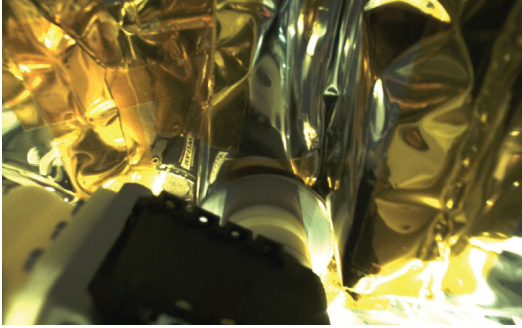


Fig. 5. Conventional teleoperation view during cutting.

unaltered tool camera image for both the left and the right eye, as shown in Fig. 5.

C. Teleoperation System Implementation

The master console of a da Vinci surgical robot (Intuitive Surgical, Sunnyvale, CA) is used to teleoperate a Universal Robots UR5 (Universal Robots, Odense, Denmark). Specifically, the operator uses the left da Vinci Master Tool Manipulator (MTM) to provide the commanded pose of the slave UR5 robot. The MTM is controlled by the da Vinci Research Kit (dVRK) open source controllers [9]. In principle, it is possible to use the da Vinci MTM to control the virtual camera, but in our implementation we use a SpaceNavigator (3Dconnexion, Munich, DE) 3D mouse, placed on the arm rest of the master console. While performing the task, the operator uses the stereo visualization system on the da Vinci master console to observe either the view from the slave robot's tool camera or the augmented virtuality view described in Section III-B.

The UR5 end-effector is mounted with a rotary cutting tool and a camera, as shown in Figs. 4, 5, 6 and 8. The cutter is

composed of a 45 mm rotary blade (Arteza, Wilmington, DE) attached to a Dynamixel MX-12W servo motor (Robotis, Lake Forest, CA). The motor is attached to a 6 axis force/torque sensor (JR3 Inc., Woodland, CA) by a 3D printed curved link. The blade is inserted on a mandrel and is secured between two concentric adapters. The overall length of the tool is 121 mm. The motor speed can be controlled in software, with a maximum no load speed of 470 RPM. Additional safety features such as maximum load and maximum torque limits are implemented to protect the hardware and the environment.

The tool camera is a lightweight 1080p High Definition device (PointGrey BlackFly, FLIR Integrated Imaging Solutions Inc., BC, Canada) mounted rigidly on the robot's end effector, oriented towards the cutting tool, as shown in Fig. 8. This approximates the expected NASA eye-in-hand configuration, where a pair of orthogonal cameras are mounted on the end-effector to provide high definition views of the tools and their interactions with the client satellite.

The teleoperation software communicates the commanded motion from the user's manipulation of the MTM to the UR5. In augmented virtuality mode, the operator provides position input. The command position of the UR5 relative to the virtual camera is set to the measured position of the MTM relative to the da Vinci stereo viewer, with a translation offset to allow the smaller MTM workspace to accommodate the larger UR5 workspace:

$$p_{cmd} = T_{vc,s} T_{m,sv} p_{mtm} + p_{off} \quad (1)$$

where p_{mtm} and p_{cmd} are the measured MTM pose and commanded UR5 pose, $T_{vc,s}$ is the transform from the UR5 to the virtual camera, $T_{m,sv}$ is the transform from the MTM to the da Vinci stereo viewer, and p_{off} is the stored offset, with orientation component fixed at identity. Operators can "clutch" to adjust the position offset at any time by depressing one of the foot pedals in the master console. As shown in Fig. 2, commands to the robot are sent as a twist derived from the desired position,

$$\begin{aligned} v_{cmd} &= \min(k_p x_{cmd}, v_{max}) \\ \omega_{cmd} &= \min(k_\theta \theta_{cmd}, \omega_{max}) \end{aligned} \quad (2)$$

where v_{cmd} and ω_{cmd} are the linear and angular velocity components of the commanded twist, x_{cmd} and θ_{cmd} are the position and orientation components of p_{cmd} , and k_p , k_θ , v_{max} , and ω_{max} are constant.

In conventional mode, because the real camera moves with the blade, it is unintuitive to use position input. We therefore

use rate input, where the MTM is servoed to a fixed position p_{fixed} and the user displaces it proportionally to the desired rate, similar to a joystick. The command twist is computed using (2) with $p_{\text{cmd}} = T_{\text{vc},s} T_{\text{m},\text{sv}}(p_{\text{fixed}} - p_{\text{mtm}})$.

The teleoperation software is aware of all virtual fixtures that have been defined in the virtual environment (see Sec. III-A.3). For this application, each virtual fixture is assumed to be a plane, but the software could be extended to support more complex fixtures. The operator may choose to use the virtual fixture as simply a visual cue to aid in aligning the blade or can press a foot pedal to “attach” to the virtual fixture. When the tool is attached to a virtual fixture, the control mode is modified to assist the operator in keeping the cutting blade parallel to the virtual fixture plane, while still allowing the operator to override this assistance if necessary (i.e., a soft virtual fixture). The soft virtual fixture is achieved by the use of non-isotropic gains. Specifically, lower gains are used for the direction perpendicular to the plane (along the z axis in Fig. 4) and for the rotations out of the plane (about the x and y axes in Fig. 4), resulting in slower motion in any direction not parallel to the virtual fixture.

Assistance in alignment of the blade to the virtual fixture is provided by an imposed force gradient that guides the blade into alignment with the virtual fixture. The force gradient is implemented by projecting the robot’s current pose into the virtual fixture plane to obtain a p_{guide} , then computing a corresponding twist as in (2) with p_{guide} replacing p_{cmd} . The resulting twist is added to v_{cmd} and ω_{cmd} . This results in the blade drifting into alignment with the virtual fixture if the operator allows the motion, but enables the operator to override the motion if necessary.

IV. EXPERIMENTS

A. Mock Satellite and MLI Hat

It is desirable to evaluate MLI cutting performance under realistic conditions, since minor details of material selection and visual appearance may significantly affect the success of execution. Thus, after consultation with NASA engineers, we constructed realistic models of the MLI hat on the Landsat 7 satellite, and mounted them on our scaled down mock satellite platform [20] shown in Fig. 6.

The natural features of the mock satellite, and their known locations in the corresponding CAD model, were used to establish registration between the robot platform and the satellite, as described in Section III-A.1. The MLI blanket covering the frame of the satellite provided a realistic and stable platform for the MLI hat.

The hats were fabricated from publicly available, non-aerospace-rated materials, but the materials were carefully selected and assembled to accurately simulate the rigidity, thickness, texture, and visual appearance of the flight-rated blanket (Fig. 7).

B. Teleoperation Pilot Study

The purpose of this pilot study is to evaluate whether the scene modeling and AV visualization makes it easier for operators to perform motions that are representative of cutting.

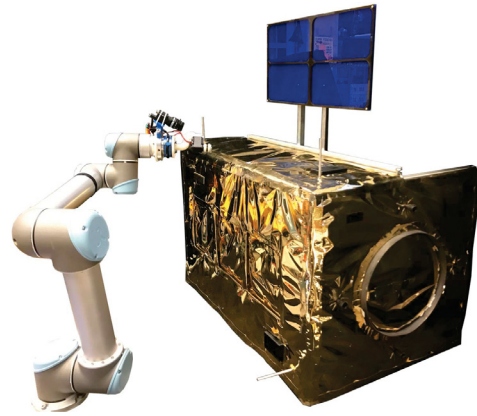


Fig. 6. Mock satellite and the UR5 robot equipped with the tool camera and the rotary cutting tool.

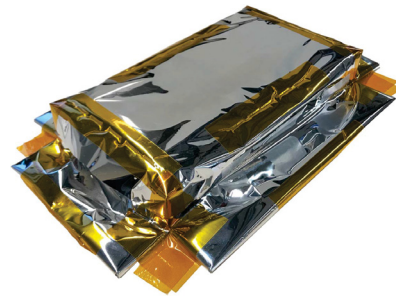


Fig. 7. Realistic mock-up of MLI ‘hat’

The operators’ performance while using a conventional teleoperation interface was compared to their performance using our proposed AV visualization interface and the virtual fixture. The cutting task was simulated by a non-destructive drawing task, using a circular crayon “blade” (Fig. 8). We placed a foam support inside the MLI hat to provide sufficient rigidity for the drawing task; this also keeps the MLI hat from sagging due to the effects of gravity in our ground-based platform.

Correctly registered virtual fixtures are enabled by the registration and scene modeling steps performed for AV visualization, therefore in our experiments the use of virtual fixtures was limited to the AV visualization tasks.

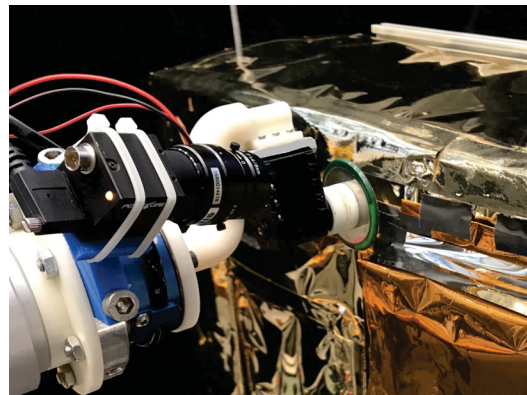


Fig. 8. Crayon blade mounted on the cutting tool during the pilot study.

TABLE I

MEAN (STANDARD DEVIATION) OF ANGULAR DEVIATIONS OF THE BLADE DURING THE ENTIRE LENGTH OF THE CUTTING TASK (DEGREES), NOT INCLUDING PERIODS WHEN THE SLAVE ROBOT WAS NOT MOVING.

	Augmented Virtuality		Conventional Teleoperation	
	Blade \angle VF Plane	Blade \angle Trajectory	Blade \angle VF Plane	Blade \angle Trajectory
User 1	8.4 (3.9)	8.3 (9.2)	6.8 (3.8)	13.6 (11.5)
User 2	2.0 (1.4)	6.5 (10.2)	11.5 (3.2)	11.5 (11.5)
User 3	3.1 (1.4)	10.1 (16.1)	9.6 (4.1)	13.8 (15.3)
User 4	7.9 (6.9)	12.9 (15.4)	17.2 (8.6)	15.0 (12.0)
User 5	3.9 (3.0)	9.8 (10.6)	20.2 (5.9)	15.6 (16.0)
User 6	4.6 (2.9)	11.5 (10.6)	19.9 (5.5)	18.0 (10.5)
User 7	3.1 (3.1)	10.1 (12.6)	22.2 (6.0)	16.1 (14.9)
All Users	4.7 (3.7)	9.9 (12.4)	15.3 (5.6)	14.8 (13.3)
z test h(p)	1 (5.7e-7)	0 (0.33)	1 (2.7e-14)	0 (0.30)

TABLE II

TASK EXECUTION TIMES (SECONDS) FOR EACH USER. THE AUGMENTED VIRTUALITY (AV) VALUES CORRESPOND TO THE SAME PROGRESS ALONG THE PATH AS THE CONVENTIONAL TELEOPERATION (CONV.) TIMES.

User	1	2	3	4	5	6	7	Mean	σ	z test h(p)
AV	178	176	396	321	357	231	344	286	90	1 (p=0.026)
Conv.	316	449	382	662	1430	600	418	608	382	1 (p<0.001)

The seven test subjects were all familiar with the design of the system, reflecting the use of skilled operators in the real-world cutting task. Operators were instructed on the use of the system and allowed to practice. For the trials with the virtual fixture, the fixture was pre-defined and always visible in the AV view. Operators were free to activate and deactivate the virtual fixture control features at will. For the conventional teleoperation trials, operators were asked to draw a straight line near the base of the MLI hat, but were not asked to follow a specific trajectory. Each operator performed the conventional teleoperation task first and the AV teleoperation task second.

For both control approaches, the operators started with the top of the MLI hat and continued smoothly to the side. When using AV, the operator also continued to the bottom of the hat but with the delayed camera image projection disabled, mimicking the expected task where one of the sides may be cut without camera visibility. In AV mode, this restriction means only that the operator has less visual feedback. However, in the conventional teleoperation mode, the task is impossible without visualization, so only the top and side are cut.

The resulting paths from all trials are shown in Fig. 9. Green lines indicate proper cutting, while other colors indicate various types of errors. While using AV mode, shown at the top, operators were able to both keep much straighter paths and keep the cutting blade properly positioned. Using the conventional teleoperation mode, operators struggled to maintain the proper drawing angle, maintain contact, and maintain a straight path. We expect this is caused by loss of situational awareness when using the narrow view from the tool camera, which is further exacerbated by the reflectivity of the MLI.

Fig. 10 and Table I show the difference in orientation

between the blade and virtual fixture plane for both modes. Similarly, these show that the majority of operators were better able to maintain the proper blade alignment using AV with virtual fixtures than using conventional teleoperation. These also show that some operators were not able to finish cutting the side using conventional teleoperation after becoming completely disoriented. The mean and standard deviation for all users in Table I is computed by averaging the individual means and variances, which assumes that each user's data is an independent random variable. A Z-test analysis of the data in Table I shows that the angular deviation of the blade with respect to the VF plane is distinct between the AV and conventional teleoperation populations with a 95% confidence level, but that the deviation of the blade with respect to the trajectory is not distinct.

Table II shows the execution times for the drawing task in the AV and conventional teleoperation modes. The time listed for both trials is the time until the blade reached the endpoint of that operator's conventional teleoperation task, which controls for both the additional bottom-side "cut" in AV mode and the fact that some operators were not able to complete both sides in the conventional teleoperation mode. For all but one operator, the AV task was completed more quickly than using conventional teleoperation. On average, the AV task was completed in less than half the time as the conventional teleoperation task. This is partially attributed to the fact that the conventional teleoperation mode did not provide visual feedback to the operators until the delayed camera feed caught up to their actions, causing them to adopt a move-and-wait approach. Using AV, operators could see the commanded position of the robot in real time, allowing them a level of feedback to perform longer stretches of the task without waiting for confirmation from the delayed camera feed. A Z-test analysis of the data in Table II shows that the AV population is distinct from the conventional teleoperation population with a 95% confidence level.

Fig. 11 shows the robot speed over time for a single representative trial. The move-and-wait strategy is clearly visible for the entire duration of the conventional teleoperation, while the augmented virtuality teleoperation had periods of sustained motion. The extra time spent waiting for the delayed visual feedback more than compensates for the time spent moving the virtual camera. An analysis of the Cartesian velocities for all trials reveals that operators using conventional teleoperation paused, for 5 or more seconds, 3.25 times more often than operators using AV ($p < 0.001$).

The reported system, including an augmented virtuality view and virtual fixtures, allowed the operators to perform the task more quickly and accurately than with conventional teleoperation. Using AV, the paths were straight with minimal gaps, while with conventional teleoperation the paths deviated significantly from a straight line and contained large gaps. Additionally, the blade angle deviation from the ideal cutting plane was minimal using AV and often extreme using conventional teleoperation. It should be noted that the effect of blade angle deviation with respect to the desired trajectory would be greater when using this system to cut MLI, as

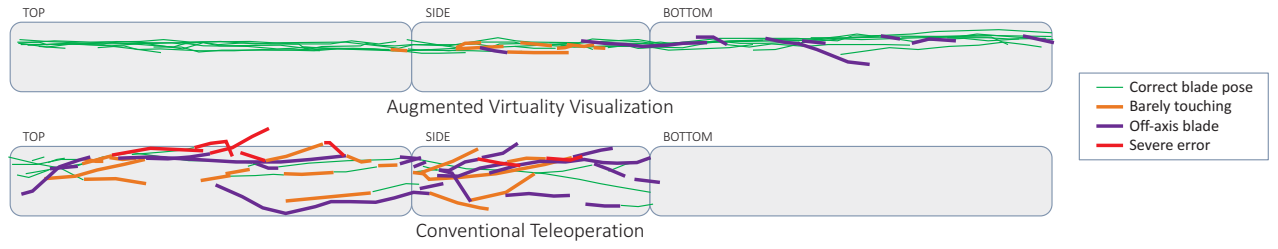


Fig. 9. Paths drawn on hat by all users; three sides for augmented virtuality and two sides for conventional teleoperation.

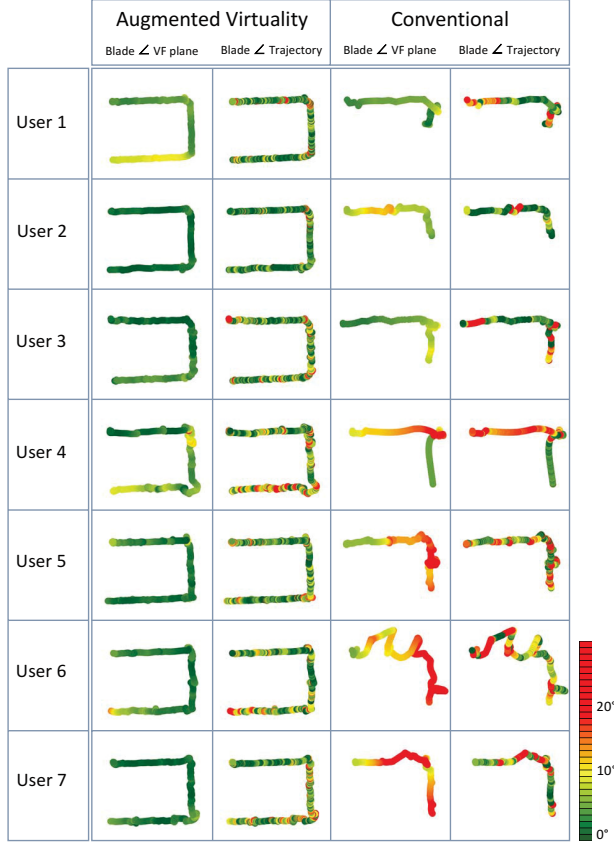


Fig. 10. Visualization of blade trajectories projected on the virtual fixture plane. Colors indicate the angular deviation of the blade from the virtual fixture plane (columns 1 and 3) and the angular deviation of the blade from the direction of motion (columns 2 and 4).

compared to the drawing task, because a skewed path may force the blade off the desired trajectory or rip the MLI.

V. DISCUSSION AND CONCLUSIONS

Due to the high risk of remote operations in space, a human operator will remain in the loop whenever feasible. Our approach, therefore, was to develop a system to improve the performance of the human operator in the challenging case of teleoperation with multi-second time delay and limited camera views. This system relies on three capabilities: (1) creation of a 3D model of the satellite that is registered with respect to the coordinate system of the remote robot, (2) visualization of the 3D model from arbitrary viewpoints, augmented by projections of the (delayed) camera feedback, and (3) motion assistance, in the form of virtual fixtures. Clearly, the latter two capabilities rely on the model created

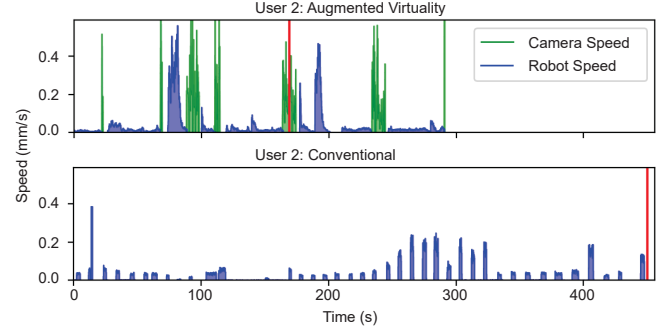


Fig. 11. Robot and camera speed over time during a typical run. In AV mode, the velocity of the slave robot is smooth and continuous, and the operator only stops moving the robot to adjust the virtual camera. In Conventional mode, the operator moves the robot a short distance at a time, then waits for the delayed video feedback; this results in a characteristic move-and-wait pattern. Red lines represent the bottom corner, which is the stopping point for the conventional teleoperation mode. As the location of the red lines indicate, in this particular run, the operator reached the bottom corner of the MLI hat in a significantly shorter time in AV mode.

by the first. The experiments demonstrate that the proposed system can significantly improve user performance for time-delayed teleoperation, compared to a baseline case with no modeling, a fixed view from the tool camera, and no motion assistance. Moreover, this augmented virtuality system enabled the human operators to reliably perform otherwise infeasible teleoperation tasks in “blind spots”, in this case, the bottom of the MLI hat, that cannot be imaged directly by the camera at the remote site. Note that our experiments did not identify the relative benefits of augmented visualization and motion assistance, which remains a topic for future studies.

In addition to teleoperation, a primary role for the human operator is to carefully monitor operations and handle unexpected conditions. For the MLI cutting task, this is important because the MLI hat is not rigid and is likely to deform and/or shift. Because the real video feedback is registered with, and projected onto, the model, the augmented virtuality interface enables the operator to detect these changes by observing the discrepancy between the projected camera image and the underlying model. For small discrepancies, the operator can compensate by adjusting the path, which is allowed by the soft virtual fixture constraints. If the discrepancy becomes large, the operator can pause the cutting operation, perform another image survey to update the model, and then resume cutting.

Our pilot studies of drawing on the MLI surface revealed several opportunities for improvement, both with the augmented virtuality and conventional interfaces. With the

augmented virtuality interface, it was sometimes challenging for the operator to obtain the desired virtual view using the 3D mouse. One simple improvement would be to allow the operator to quickly select from a set of views that are pre-defined with respect to the cutting blade; for example, views from above or in front of the blade wheel. This set of pre-defined views could be augmented by the operator, who could “save” a particular view and then recall it later. For the conventional interface, it would have been helpful to include a predictive display [3] to give the operator a better sense of how the commanded robot motion compared to the currently visualized position. NASA’s planned servicing spacecraft is also equipped with several situational awareness (SA) cameras mounted on the base platform (deck) of the robot arms. While the remote locations of the SA cameras prevent them from providing high quality close-up imaging of the immediate surrounding of the robotic tools, operator performance during our experiments would likely have benefited from having additional SA camera views available, especially during conventional teleoperation. Another limitation of the reported experiments was the use of a fixed round-trip telemetry delay of 5 seconds; in an actual mission, this delay is expected to vary by several seconds, which would further degrade teleoperation performance.

We are currently evaluating the system for actual MLI hat cutting experiments, using trained NASA robot operators, and will report the results in subsequent publications. For those experiments, we developed a more representative conventional teleoperation interface that uses a keyboard and GUI instead of the da Vinci master console. This interface also includes the predictive display and deck cameras that were absent in the experiments reported here.

ACKNOWLEDGMENTS

This work was supported by NASA NNG15CR66C. Ryan Reedy received support from the NSF REU program via EEC-1460674. We thank Billy Gallagher, Brian Roberts, Erich Shultze, and their colleagues at NASA GSFC for technical discussions and support. Zihan Chen and Paul Wilkening assisted with software development. Shuyang Chen and Jiajun Li designed and built the camera attachment for the robot and Jiajun Li assisted with the cutting motor hardware and software. Shane Gilligan-Steinberg assisted with the experimental setup. The da Vinci Research Kit (dVRK) is supported by NSF NRI-1637789.

REFERENCES

- [1] T. Abuhamdia and J. Rosen, “Constant visual and haptic time delays in simulated bilateral teleoperation: Quantifying the human operator performance,” *Presence: Teleoperators and Virtual Environments*, vol. 22, no. 4, pp. 271–290, 2013.
- [2] A. K. Bejczy, W. S. Kim, and S. C. Venema, “The phantom robot: predictive displays for teleoperation with time delay,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 1990, pp. 546–551.
- [3] A. Bejczy and W. Kim, “Predictive displays and shared compliance control for time-delayed telemanipulation,” in *IEEE Intl. Workshop on Intelligent Robots and Systems (IROS)*, 1990, pp. 407–412.
- [4] M. Draeos, B. Keller, C. Toth, A. Kuo, K. Hauser, and J. Izatt, “Teleoperating robots from arbitrary viewpoints in surgical contexts,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, Sept. 2017, pp. 2549–2555.
- [5] W. R. Ferrell, “Delayed force feedback,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 8, no. 5, pp. 449–455, 1966.
- [6] J. Funda, T. S. Lindsay, and R. P. Paul, “Teleprogramming: Toward delay-invariant remote manipulation,” *Presence: Teleoperators & Virtual Environments*, vol. 1, no. 1, pp. 29–44, 1992.
- [7] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, “Sensor-based space robotics-ROTEX and its telerobotic features,” *IEEE Trans. on Robotics and Automation*, vol. 9, no. 5, pp. 649–663, Oct 1993.
- [8] I. Ince, K. Bryant, and T. Brooks, “Virtuality and reality: a video/graphics environment for teleoperation,” in *IEEE Intl. Conf. on Systems, Man, and Cybernetics (SMC)*, 1991, pp. 1083–1089.
- [9] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, “An open-source research kit for the da Vinci® surgical system,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 6434–6439.
- [10] D. Koppel, Y.-F. Wang, and H. Lee, “Image-based rendering and modeling in video-endoscopy,” in *IEEE Intl. Symp. on Biomedical Imaging: Nano to Macro*, 2004, pp. 269–272.
- [11] Y. Koreeda, S. Obata, Y. Nishio, S. Miura, Y. Kobayashi, K. Kawamura, R. Souzaki, S. Ieiri, M. Hashizume, and M. G. Fujie, “Development and testing of an endoscopic pseudo-viewpoint alternating system,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 10, no. 5, pp. 619–628, 2015.
- [12] M. A. Menchaca-Brandan, A. M. Liu, C. M. Oman, and A. Natapoff, “Influence of perspective-taking and mental rotation abilities in space teleoperation,” in *ACM/IEEE Intl. Conf. on Human-Robot Interaction (HRI)*, 2007, pp. 271–278.
- [13] P. Milgram and F. Kishino, “A taxonomy of mixed reality visual displays,” *IEICE Trans. on Information and Systems*, vol. 77, no. 12, pp. 1321–1329, 1994.
- [14] P. Mitra and G. Niemeyer, “Model-mediated telemanipulation,” *Intl. Journal of Robotics Research*, vol. 27, no. 2, pp. 253–262, Feb. 2008.
- [15] NASA GSFC, “Restore-L Robotic Servicing Mission.” [Online]. Available: <https://sspd.gsfc.nasa.gov/restore-L.html>
- [16] L. B. Rosenberg, “Virtual fixtures: Perceptual tools for telerobotic manipulation,” in *IEEE Virtual Reality Annual Intl. Symp.*, Seattle, WA, Sept. 1993, pp. 76–82.
- [17] T. Sheridan, “Space teleoperation through time delay: review and prognosis,” *IEEE Trans. on Robotics and Automation*, vol. 9, no. 5, pp. 592–606, Oct. 1993.
- [18] E. H. Spain, “Stereoscopic versus orthogonal view displays for performance of a remote manipulation task,” in *SPIE, Stereoscopic Displays and Applications III*, Feb. 1991, pp. 103–110.
- [19] R. Y. Tsai and R. K. Lenz, “Real time versatile robotics hand/eye calibration using 3D machine vision,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 1988, pp. 554–561.
- [20] B. Vagvolgyi, W. Niu, Z. Chen, P. Wilkening, and P. Kazanzides, “Augmented virtuality for model-based teleoperation,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, Sept. 2017, pp. 3826–3833.
- [21] S. Vozar, Z. Chen, P. Kazanzides, and L. L. Whitcomb, “Preliminary study of virtual nonholonomic constraints for time-delayed teleoperation,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2015, pp. 4244–4250.
- [22] S. Vozar, S. Leonard, P. Kazanzides, and L. L. Whitcomb, “Experimental evaluation of force control for virtual-fixture-assisted teleoperation for on-orbit manipulation of satellite thermal blanket insulation,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 4424–4431.
- [23] T. Xia, S. Léonard, A. Deguet, L. Whitcomb, and P. Kazanzides, “Augmented reality environment with virtual fixtures for robotic telemanipulation in space,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2012, pp. 5059–5064.
- [24] T. Xia, S. Léonard, I. Kandaswamy, A. Blank, L. Whitcomb, and P. Kazanzides, “Model-based telerobotic control with virtual fixtures for satellite servicing tasks,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2013, pp. 1479–1484.
- [25] E. Yang and M. Dorneich, “The emotional, cognitive, physiological, and performance effects of variable time delay in robotic teleoperation,” *Intl. J. of Social Robotics*, vol. 9, no. 4, pp. 491–508, Sep 2017.
- [26] W.-K. Yoon, T. Goshozono, H. Kawabe, M. Kinami, Y. Tsumaki, M. Uchiyama, M. Oda, and T. Doi, “Model-based space robot teleoperation of ETS-VII manipulator,” *IEEE Trans. on Robotics and Automation*, vol. 20, no. 3, pp. 602–612, June 2004.